链滴

设计模式（工厂方法）- 接口多态选择

作者：Pleuvoir
原文链接：https://ld246.com/article/1522470408980
来源网站：链滴

工厂模式（Factory Pattern）是 Java 中最常用的设计模式之一。这种类型的设计模式属于创建型模，在工厂模式中，我们在创建对象时不会对客户端暴露创建逻辑，并且是通过使用一个共同的接口来向新创建的对象。

代码示例：

接口

```
public interface ChannelService {

    void wechat();

    void alipay();
}
```

具体的接口实现类

```
public class AchannelService implements ChannelService {

    @Override
    public void wechat() {
        System.out.println("A渠道微信");
    }

    @Override
    public void alipay() {
        System.out.println("A渠道支付宝");
    }
```

```
}

public class BchannelService implements ChannelService {

    @Override
    public void wechat() {
        System.out.println("B渠道微信");
    }

    @Override
    public void alipay() {
        System.out.println("B渠道支付宝");
    }
}
```

工厂方法

```
public class ChannelServiceFactory {

    private static ChannelService achannelService = new AchannelService();
    private static ChannelService bchannelService = new BchannelService();

    public static ChannelService route(String channelCode) {
        if (channelCode.equals("A")) {
            return achannelService;
        } else if (channelCode.equals("B")) {
            return bchannelService;
        }
        throw new IllegalArgumentException();
    }
}
```

测试

```
    public static void main(String[] args) {
        ChannelServiceFactory.route("A").wechat();
        ChannelServiceFactory.route("B").wechat();
    }
```

结论

我们明确地计划不同条件下创建不同实例时，选用工厂模式，只需要从工厂中获取服务，调用方不用心具体是哪种实现

